Building Data Management Systems for DOE Science

Reagan W. Moore
Wayne Schroeder
San Diego Supercomputer Center

June 30, 2004

Scientific simulations now generate massive amounts of data – millions of files and hundreds of terabytes of data. Scientific teams also generate massive amounts of observational data – billions of files and petabytes of data. The management of data has become an essential capability for the conduct of science.

Data management systems come in multiple flavors:

- Real-time data organization and preservation. Examples are sensor systems and web portals.
- Collection building. Examples are the additional of a descriptive context to a set of files to understand the relationships between the files. An example is organizing all of the output from a simulation code into a collection, tracking the changes in input parameters for each simulation output. Queries can then be made over the parameter space to identify relevant files.
- Data sharing. Examples are data grids that provide a global name space for accessing files, support replication of files between file systems and archives, and provide access through user-preferred access mechanisms.
- Data publication. Examples are digital libraries that assemble a collection, and support browsing, query, and display services for interactively examining the context associated with the files.
- Data preservation. Examples are persistent archives that manage technology evolution through the migration of collections onto new systems.
- Data analysis. Examples are dataflow systems that extract data from collections, process the collection subset, and then register the results back into the collection.

The critical component of all of these systems is the association of a context with data. Files that do not have an associated context are useless. Only the creator of such a file understands the relevance, how the file was created, and the relationship of the file to other data. The context is expressed as properties associated with each file. The properties can be named, and organized in standard metadata schema for a discipline. Examples of properties include the assignment of a global persistent identifier to the data, descriptive metadata, state information that is generated by operations on the data, structural information, and provenance information describing how the data was generated. Instances of properties are implemented as metadata attribute values and stored in a catalog. The catalog in turn can be implemented as tables in a database. Relationships between the properties can be described through ontologies. The expectation is that through access to a data context, discovery of relevant data can be automated, access to data can be automated, and manipulation of data can be automated.

A driving motivation for the development of data management systems is the evolution of science into a collaborative enterprise that spans administrative domains, that spans multiple institutions, and that even spans disciplines. The conduct of science requires the ability to share, publish, preserve, and analyze massive data collections that are themselves geographically distributed across heterogeneous storage systems.

The mechanisms required for management of data are distinct from mechanisms for accessing data. Management of data requires the ability to maintain consistency between the collection context (the metadata) and the collection content (the data files). Operations on files create state information that needs to be saved as part of the collection context. Examples of state information include the physical location of a file after it has been moved, the existence of replicas of the files, audit trails that track accesses to the data, checksums that are used to assert integrity of the data, access control lists that are used to authorize manipulation, new descriptive metadata that is generated after analysis of the files, etc. Essentially every operation on data generates information that can be used to define the context (meaning and relevance) of each file.

Access mechanisms provide a way to discover and retrieve data. Access mechanisms are characterized by their diversity, and include C library calls, C++ library calls, Java classes, Perl load libraries, Python load libraries, unix shell commands, Open Archives Initiative for harvesting metadata, Web browsers, Windows browsers, Java browsers, Storage Resource Manager, Web Service Definition Language, I/O system call redirection, I/O system call load libraries, etc.. Researchers desire the ability to use their preferred access mechanism when manipulating data.

Data grid technology provides the core capabilities required to support the wide variety of data management systems, including the sharing, publication, preservation, and analysis of data. Data grids provide the interoperability mechanisms for accessing heterogeneous storage systems, the metadata catalogs for managing state information and descriptive metadata, the consistency mechanisms, interoperability between authentication environments, location independent authorization mechanisms, and access interoperability mechanisms needed to support the wide variety of user interfaces.

Data grids are being used to support high energy physics collections, earth science collections, bio-informatics collections, education collections, astronomy collections, and many other disciplines in the NSF community. The DOE Science community can greatly improve access to their research results by implementing data grids between the DOE laboratories that support open science research. Of even greater significance would be the federation of DOE data grids with data grids supported by other federal agencies. The sharing of data between federal agencies would accelerate scientific progress, and improve the ability of researchers to access relevant material needed to promote understanding of basic scientific processes. The technology needed to support data grid federation is now available through the SDSC Storage Resource Broker version 3 software.